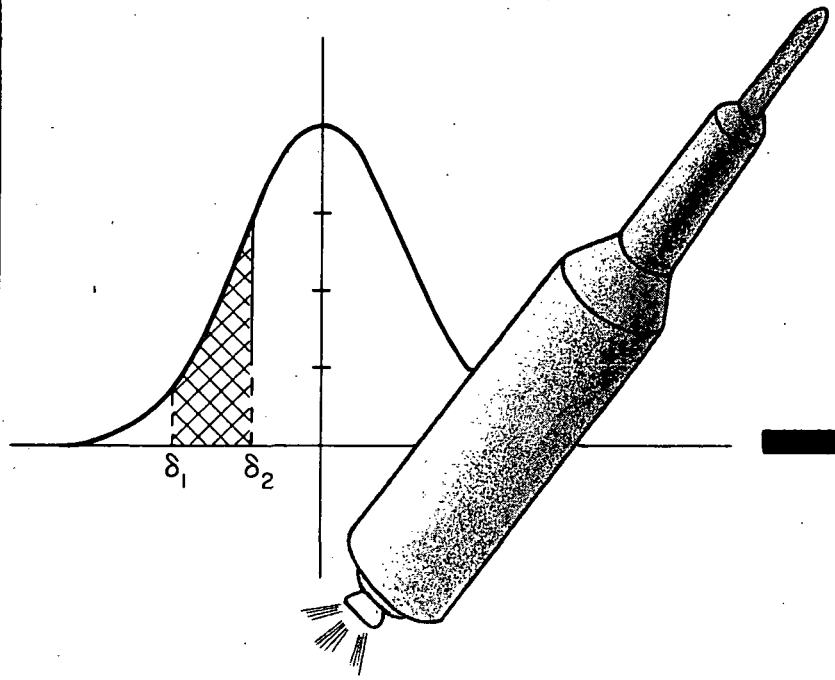
A USER'S GUIDE FOR A GENERALIZED INTERPLANETARY TRAJECTORY GENERATION PROGRAM

June 17, 1972



TEXAS CENTER FOR RESEARCH
3100 PERRY LANE, AUSTIN, TEXAS

A USER'S GUIDE FOR A GENERALIZED INTERPLANETARY TRAJECTORY GENERATION PROGRAM

June 17, 1972

Texas Center for Research

3100 Perry Lane

Austin, Texas 78731

A USER'S GUIDE FOR A GENERALIZED INTERPLANETARY TRAJECTORY COMPUTATION PROGRAM

by Texas Center for Research

1.0 SUMMARY

The analysis, structure, and capability of a generalized precision interplanetary trajectory computation program are discussed, with emphasis being placed on the description of input and output. Sample cases showing input and output information are included.

2.0 INTRODUCTION

This report describes a computer program which was developed by the Texas Center for Research of Austin, Texas, for computing precision interplanetary trajectories. The program uses numerical optimization techniques and a variation of Newton's method to produce trajectory iterates which converge to an approximate n-body interplanetary trajectory. This approximate trajectory is then used with a perturbation procedure to produce the precision n-body interplanetary trajectory.

This procedure was first tested on earth-moon trajectories where it was used for a published paper (ref. 1) and several reports (refs. 2, 3).

The procedure developed in this study has been used to compute two-dimensional and three-dimensional earth-moon trajectories and two- and three-dimensional interplanenery n-body trajectories using circular planetary orbits, elliptic planetary orbits, and planetary orbits specified by the JPL Analytical Ephemeris. In addition, trajectories from earth to Venus to a heliocentric radius vector with a Δv at Venus have been computed. The program was then used in an iterative mode to minimize the Δv at Venus in order to simulate the free-fall fly-by. The modifications necessary to provide this capability will be made in the program and an addendum to the User's Manual will be provided.

This program, possibly with some minor modifications, will fill a variety of needs for precision interplanetary trajectories. This report will not cover all possible modifications of the program, but will indicate the types of modifications which could be made and will outline the potential uses of various modified versions of the program.

3.0 PROBLEM ANALYSIS

The trajectory computation problem for free-fall trajectories can be stated as follows:

Find the set of initial velocities $v(t_i)$ such that a particle subject to the equations of motion

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= F(x,t) \end{aligned} \tag{1}$$

moves from a known initial point (x_i,t_i) to a known final point (x_f,t_f). This is a classical two-point boundary value problem. It is not an optimal control problem as there are no controls in this formulation.

The details of the derivation will not be covered here. Only the important equations and how the technique works will be presented.

Primary Convergence Algorithm

Rewrite the equations as

$$\begin{aligned} \dot{x} &= u(t) \\ \dot{\lambda} &= F(x,t) \end{aligned} \tag{2}$$

with initial conditions $\lambda(t_i) = u(t_i)$. Clearly, we want to change the system in such a way that $u(t)$ and $\lambda(t)$ approach the same values, i.e., $u(t)$. Note that if a $u(t)$ history is guessed (a control program in the optimal control vocabulary) then a position history, $x(t)$, and a history of the variables $\lambda(t)$ can be obtained.

The procedure used to drive $u(t)$ and $\lambda(t)$ to the same values requires that both of equations (2) above be linearized. The new values for $u(t)$ and $\lambda(t)$ are given by

$$\begin{aligned} u_{new}(t) &= u(t) + \delta u(t) \\ \lambda_{new}(t) &= \lambda(t) + \delta\lambda(t) \end{aligned} \tag{3}$$

Setting $u_{new}(t)$ equal to $\lambda_{new}(t)$, one obtains

$$\delta u(t) = \delta\lambda(t) + \lambda(t) - u(t) \tag{4}$$

where $u(t)$ and $\lambda(t)$ are evaluated on the previous iterate.

In order to control stepsize and to keep consecutive iterates close enough to each other so that linearization is valid, the scalar variable $P(0 \leq P \leq 1)$ is introduced into equation (4). Equation (4) is modified in the manner shown below

$$\delta u(t) = \delta\lambda(t) + P[\lambda(t) - u(t)] \tag{5}$$

Linearization of equations (2) leads to

$$\begin{aligned} \delta\dot{x} &= u(t) \\ \delta\dot{\lambda} &= F_x \delta x(t) \end{aligned} \tag{6}$$

where F_x is a 3 x 3 matrix of partial derivatives of the F_x's with respect to the position coordinates. Substitution for δu in equation (6) from equation (5) leads to

$$\begin{aligned} \delta\dot{x} &= \delta\lambda(t) + P[\lambda(t) - u(t)] \\ \delta\dot{\lambda} &= F_x \delta x(t) \end{aligned} \tag{7}$$

It is now assumed that both $\delta x(t)$ and $\delta\lambda(t)$ are linear functions of the initial values of δx_i and $\delta\lambda_i$. Since $\delta x_i = 0$, it is possible to write

$$\begin{aligned} \delta x(t) &= A(t)\,\delta\lambda_i + M(t) \\ \delta\lambda(t) &= B(t)\,\delta\lambda_i + N(t) \end{aligned} \tag{8}$$

where A and B are 3 x 3 time-dependent matrices and M(t) and N(t) are time-dependent 3-vectors. The conditions which define A, B, M and N will be determined below.

From the conditions $\delta x(t_i) = 0$ and $\delta\lambda(t_i) = \delta\lambda_i$, the initial conditions on A, B, M and N can be defined; i.e.,

$$\begin{aligned} A(t_i) &= 0 \\ B(t_i) &= I \quad \text{(the 3 x 3 identity matrix)} \\ M(t_i) &= 0 \\ N(t_i) &= 0 \end{aligned} \tag{9}$$

Taking derivatives of equations (8) and substitution of these results plus those of equations (8) into equations (7) leads to the relations

$$\begin{aligned} \dot{A}\,\delta\lambda_i + \dot{M} &= B\,\delta\lambda_i + N + P(\lambda - u) \\ \dot{B}\,\delta\lambda_i + \dot{N} &= F_x A\,\delta\lambda_i + F_x M \end{aligned} \tag{10}$$

Thus, the conditions defining A, B, M and N are

$$\begin{aligned} \dot{A} &= B & A(t_i) &= 0 \\ \dot{B} &= F_x A & B(t_i) &= I \\ \dot{M} &= N + P(\lambda - u) & M(t_i) &= 0 \\ \dot{N} &= F_x M & N(t_i) &= 0 \end{aligned} \tag{11}$$

From these relations, A(t), B(t), M(t), and N(t) can be determined. Then,

$$\delta\lambda_i = A^{-1}(t_f)\,[\delta x(t_f) - M(t_f)] \tag{12}$$

and

$$\delta\lambda(t) = B(t)A^{-1}(t_f)[\delta x(t_f) - M(t_f)] + N(t) \tag{13}$$

Also,

$$\delta u(t) = B(t)A^{-1}(t_f)[\delta x(t_f) - M(t_f)] + N(t) + P[\lambda(t) - u(t)] \tag{14}$$

The computational procedure is as follows:

1. Guess $u(t)$
2. Integrate the equations below from t_i to t_f with the indicated initial conditions:

$$\begin{aligned} \dot{x} &= u(t) & x(t_i) &= x_i \text{ , given} \\ \dot{\lambda} &= F(x,t) & \lambda(t_i) &= u(t_i) \\ \dot{A} &= B & A(t_i) &= 0 \\ \dot{B} &= F_x A & B(t_i) &= I \\ \dot{M} &= N + P(\lambda - u) & M(t_i) &= 0 \\ \dot{N} &= F_x M & N(t_i) &= 0 \end{aligned} \tag{15}$$

3. Using the specified values for x_f and the integrated values of $x(t_f)$, calculate

$$\delta x_f = x_f - x(t_f) \tag{16}$$

4. Next, calculate

$$\delta\lambda_i = A^{-1}(t_f)[\delta x_f - M_f] \text{ ,} \tag{17}$$

and

$$\delta u(t) = B(t)A^{-1}(t_f)[\delta x_f - M_f] + N(t) + P(\lambda(t) - u(t)] \tag{18}$$

5. Now set

$$\begin{aligned} \lambda_{i_{new}} &= \lambda_i + \delta\lambda_i \\ u_{new}(t) &= u(t) + \delta u(t) \end{aligned} \tag{19}$$

6. Return to step 2 and repeat the process until $\lambda(t)$ is sufficiently close to $u(t)$.

Once the process described in steps 1 through 6 above converges, the values $(x(t_i),\lambda_i)$ are used as initial conditions $(v(t_i) = \lambda_i)$ in the integration

$$\dot{x} = v \qquad x_i \text{ specified}$$

$$\dot{v} = F(x,t) \qquad v(t_i) = \lambda_i$$

This integration is employed in a standard perturbation procedure to refine the initial values $v(t_i)$ (i.e., to refine λ_i). The perturbation procedure normally converges in three or four iterates. Note that the standard perturbation algorithm is embedded in the computational algorithm being used. As $\lambda(t) \rightarrow u(t)$, the term $P(\lambda - u)$ goes to zero and M and N become identically zero. When this occurs, equation (12) becomes the standard equation for $\delta\lambda_i$ from the perturbation algorithm and equations (13) and (14) become the same.

4.0 PROGRAM STRUCTURE

The n-body interplanetary trajectory program uses four basic groups of subroutines. The functions of the four groups of subroutines, and the individual subroutine names are listed in Table I.

The MAIN Program calls RDATA for input (see Table II for input structure) and then converts the planetocentric equatorial inputs to heliocentric ecliptic coordinates using subroutines COORDS and EPHEMP. Next MAIN calls subroutine DRIVER which controls the program from this point forward.

Subroutine DRIVER sets up the initial conditions for the first iteration and either generates a $u(t)$ history from a heliocentric trajectory patched to hyperbolic orbits near the planets (subroutine VGUESS) or generates a $u(t)$ history by integrating a guessed set of initial velocities for the transfer in the n-body solar system employing RK 3(4).

Once the initial u(t) history is known, DRIVER controls the iteration described under PROGRAM ANALYSIS until u(t) and $\lambda(t)$ match sufficiently well to allow the shift to a standard perturbation procedure for final convergence. This iteration process employs RK 3(4) for integration and SPLINE for interpolation.

When the shift is made to the perturbation procedure, the RK 7(8) integrator is employed. Both RK 3(4) and RK 7(8) obtain derivatives from subroutine DERIV.

Slightly different derivatives are called for by each procedure, so DERIV contains a branch. Subroutine DERIV uses subroutine EPHEMP and SPLINE to aid in the generation of the required derivatives.

5.0 PROGRAM CAPABILITY

The program possesses four options which may be used as described below. The option chosen is indicated by the input parameter NØPT (NØPT=1,2,3, or 4). The options and their uses are described below.

Option 1 (NØPT=1)

When operating in this mode, the program finds the n-body trajectory which goes from (t_i,x_i) to (t_f,x_f) where x_i is the specified initial position vector, t_i is the initial time, x_f is the specified final position vector, and t_f is the final time.

The initial and final position vectors are normally input in kilometers in planetocentric equatorial coordinates with x_i being measured in a system centered at planet NP(1) and x_f measured in a system centered at planet NP(2). The planetary indices used are Mercury=1, Venus=2, Earth=3, etc. If it is desired to input either x_i or x_f in heliocentric ecliptic coordinates in units of AU, the corresponding planetary index, NP(1) or NP(2), should be set to 10.

For this option, the program converts the initial and final position vectors into heliocentric ecliptic coordinates

(if necessary), and then obtains an initial velocity history from a Lambert solution to start the first order iteration process.

For NØPT=1 runs, the required input data is:

TP(1)	t_i ,	the initial time (J.D.)
TP(2)	t_f ,	the final time (J.D.)
TP(3)		unspecified
XI(1)		the initial position vector in km. if
XI(2)	x_i ,	NP(1) $\leq$ 9 in AU if NP(1) $\geq$ 10
XI(3)		
XF(1)		the final position vector in km if
XF(2)	x_f ,	NP(1) $\leq$ 9 in AU if NP(1) $\geq$ 10
XF(3)		
VI(1)		unspecified
VI(2)		
VI(3)		
VF(1)		unspecified
VF(2)		
VF(3)		
NP(1)		initial planet index
NP(2)		final planet index
NØPT		integer, =1
NPRT		integer, print key NPT=0, do not print last iterate NPRT=1, print every NSPRTth integration steps on last iterate (converged iterate). NPRT=-1, print every NSPRTth tion on every iterate.
NSPRT		integer, see NPRT

Options 2 and 3 (NØPT=2/NØPT=3)

These options are available in order to allow a run which has been terminated prematurely (due to time limit, for example) to be restarted at the last iterate computed. If the program was still operating in the first order iteration mode when the termination occurred, NØPT=2 should be used. If, however, the shift to the perturbation method has already occurred, NØPT=3 should be used.

In either case, the initial position vector x_i (XI) and the initial velocity vector v_i (VI) will be in heliocentric ecliptic coordinates in units of AU and AU/year. These values should be used directly as input for the NØPT=2/NØPT=3 options. For these options t_i TP(1) and t_f TP(2) as in NØPT=1.

For NØPT=2/NØPT=3 runs, the required input data is:

TP(1)	t_i ,	initial time
TP(2)	t_f ,	final time
TP(3)		unspecified
XI(1)		initial position vector in AU
XI(2)	x_i ,	
XI(3)		
XF(1)		final position vector in AU
XF(2)	x_f ,	
XF(3)		
VI(1)		initial velocity vector in AU/yr.
VI(2)	v_i ,	
VI(3)		
VF(1)		unspecified
VF(2)		
VF(3)		
NP(1)		unspecified (can be specified but will not be used)
NP(2)		
NØPT		integer, =2 or 3
NPRT		as in NØPT=1 option
NSPRT		

Option 4 (NØPT=4)

The purpose of this option is to enable the user to compute a trajectory which goes from (x_i, t_i) through a point (x_m, t_m) to a point (x_f, t_v) with a velocity impulse, ΔV , at (x_m, t_m) and then, by allowing the point x_m to vary, make the ΔV at (x_m, t_m) as small as possible. This option uses information obtained from two NØPT=1 runs as input data.

In order to obtain the information required for input for the NØPT=4 option, the user must set up an NØPT=1 run from (x_i, t_i) to (x_m, t_m) and get the converged initial position

and velocity components at t_i (in heliocentric ecliptic coordinates). Then he must set up an NØPT=1 run from (x_m, t_m) to (x_f, t_f) and get the converged final position and velocity components at t_f (in heliocentric ecliptic coordinates). These values are then used as XI, VI, XF, and VF (all 3-vectors) in the input for the NØPT=4 run. The times are input in the following manner: t_i TP(1), t_m TP(2), t_f TP(3). The quantities NP(1) and NP(2) do not need to be specified for an NØPT=4 run.

For NØPT=4 runs, the required input is:

TP(1)	t_i ,	initial time (J.D.)
TP(2)	t_m ,	fly-by time (J.D.)
TP(3)	t_f ,	final time (J.D.)
XI(1)		initial position vector in AU (from NØPT=1 run)
XI(2)	x_i ,	
XI(3)		
XF(1)		final position vector in AU (from NØPT=1 run)
XF(2)	x_f ,	
XF(3)		
VI(1)		initial velocity vector in AU/yr. (from NØPT=1 run)
VI(2)	v_i ,	
VI(3)		
VF(1)		final velocity vector in AU/yr. (from NØPT=1 run)
VF(2)	v_f ,	
VF(3)		
NP(1)		unspecified
NP(2)		
NØPT		integer, =4
NPRT		as in NØPT=1
NSPRT		

TABLE I

Basic Subroutine Groupings
for the
Interplanetary Trajectory Generation Program

1. Control Routines
   - MAIN
   - DRIVER

2. Numerical Integration Related Routines
   - RK34CØ
   - RK34
   - RK78CØ
   - RK78
   - DERIV
   - VGUESS
   - SPLINE

3. Celestial Mechanics Routines
   - CØNIC
   - ØETØRC
   - EPHEMP
   - TRFM
   - CØØRDS
   - EULER
   - DLMBRT
   - UVFNS

4. Vector Manipulation Routines
   - UNIT
   - DØT
   - VMAG
   - CRØSS

TABLE II

Description of Input

Variables	Description
TP(1),TP(2), TP(3) Units (Julian days) Double Precision	Times in Julian days as required for the various problem options
XI(1),XI(2),XI(3) Units (km or AU) Double Precision	Initial position vector for NØPT=1, XI is expressed in planetocentric equatorial coordinates at planet NP(1) in kilometers for NØPT=2,3, or 4, XI is expressed in heliocentric ecliptic coordinates in A.U.
XF(1),XF(2),XF(3) Units (km or AU) Double Precision	Final position vector (see XI for details on units for various values of NØPT)
VI(1),VI(2),VI(3) Units (AU/yr) Double Precision	Initial velocity vector in heliocentric ecliptic coordinates in AU/yr. (Needed only for NØPT=2,3, or 4)
VF(1),VF(2),VF(3) Units (AU/yr) Double Precision	Final velocity vector in heliocentric ecliptic coordinates in AU/yr. (Needed only for NØPT=4)
NP(1),NP(2) Integers	Planet index denoting departure planet and target planet in system where Mercury=1, Venus=2, etc.
NØPT Integer	Option choice key, NØPT=1,2,3, or 4.
NPRT Integer	Print option key NPRT=-1 Print every NSPRTth integration step on every iterate NPRT=0 Print only summary information--do not print at intervals on any iterate NPRT=1 Print every NSPRTth integration step on the last (converged) iterate only
NSPRT Integer	The number of integration steps between prints if NPRT≠0. (see NPRT)

For more detail on input, see Section 5.0, PROGRAM CAPABILITY.

The data is input using the system routine RDATA. The call to RDATA is

```
CALL RDATA(TP,XI,XF,VI,VF,NP,NØPT,NPRT,NSPRT)
```

TABLE III

Description of Output

Output Quantity	Description
ITERATION (Integer)	A counter which shows the current number of iterations used during any one mode of operation
FINAL TIME Units (years)	Elapsed time during the current iteration from the initial time
INTEGRATION STEPS (Integer)	Total number of steps taken by the integrator to complete the current iteration
XI Units (AU)	Initial position in heliocentric ecliptic coordinates (3 components)
XF Units (AU)	Final integrated position in heliocentric ecliptic coordinates (3 components)
VI Units (AU/yr)	Initial velocity (3 components)
VF Units (AU/yr)	Final integrated velocity (3 components)
DELXF Units (AU)	Terminal position miss defined as difference between input final position and integrated final position (3 components)
DUMAX Units (AU/yr)	Maximum change in the control velocity history for the next iteration of this mode 1 operation. This quantity is meaningless during modes 2 and 3 operation (3 components)
DELVI Units (AU/yr)	Change in the initial velocity for the next iteration (3 components)
MINIMUM APPROACH DISTANCES Units (AU)	The minimum radial distance obtained anytime during the current iteration from each of the solar system members listed
FINAL APPROACH DISTANCES Units (AU)	The final radial distance obtained during the current iteration from each of the solar system members listed

XI=	-.98247945670+00	-.18206340106+00	.941741311249-05
XF=	.34428633700+00	.63400365941+00	-.10819241857-01

FRACTION OF PLANETS GRAVITY USED= .010

ITERATION= 1	FINAL TIME=	.4790231348+00	INTEGRATION STEPS= 219
XI =	-.98247945670+00	-.18206340106+00	.941741311249-05
XF =	.34428633700+00	.63400365941+00	-.10819241857-01
VI =	.86151100951+00	-.56247070646+01	.10971254959+00
VF =	-.66633265871+01	.44379032209+01	-.95836137163-01
DELAF=	.25709412503-07	-.13205430697+06	.25950039847+08
DUMAA=	.48859599039D-01	.40723237797D-01	.54660794108D-02
DELVI=	.48859599039D-01	.31645121225D-01	-.54660794108D-02

MINIMUM APPPOACH DISTANCES

SUN	.69162867181+00
MERCUR	.35996432452+00
VENUS	.42731353153+04
EARTH	.20067801561+04
MARS	.79485920176+00
JUPITE	.45976046078+01

FINAL APPPOACH DISTANCES

SUN	.72153362914+00
MERCUR	.11848954379+01
VENUS	.42731353153+04
EARTH	.92491001412+00
MARS	.21942837170+01
JUPITE	.57131348305+01

FRACTION OF PLANETS GRAVITY USED= .010

ITERATION= 2	FINAL TIME=	.4790231348+00	INTEGRATION STEPS= 219
XI =	-.98247945670+00	-.18206340106+00	.941741311249-05

TABLE IV
Sample Output

XF =	.3442669057460+00	.6340021330160+00	-.1081921069720-01
VI =	.9103730684510+00	-.5593061943470+01	.1042264701810+00
VF =	-.6568793530020+01	.4485066332740+01	-.1040423429680+00
DELXF=	-.5430296908990-06	.1394338729180-05	-.2856516693020-07
DUMAX=	.5758622827640-01	.4521275139640-01	.6561721184130-02
DELVI=	.5758622827640-01	.3584780025250-01	-.6561721184130-02

MINIMUM APPPOACH DISTANCES

SUN	.6915939712910+00
MERCUR	.3596905330840+00
VENUS	.4381648265670-04
EARTH	.2282371956860-04
MARS	.7948592017610+00
JUPITE	.4597604807810+01

FINAL APPPOACH DISTANCES

SUN	.7215325588350+00
MERCUR	.1184869329600+01
VENUS	.4381648265670-04
EARTH	.9249085377660+00
MARS	.2194282465000+01
JUPITE	.5713135460310+01

FRACTION OF PLANETS GRAVITY USED= .010

ITERATION= 3 FINAL TIME= .4790231348+00 INTEGRATION STEPS= 207

XI =	-.9824794567040+00	-.1820634010866+00	.9417413112490-05
XF =	.3442871352020+00	.6340008642310+00	-.1081918561120-01
VI =	.9679568368270+00	-.5557214194220+01	.9764474869730-01
VF =	-.6509154561230+01	.4534880615010+01	-.1126803166910+00
DELXF=	-.7724852126400-06	.2663124057690-05	-.5365111338120-07
DUMAX=	.3233199605300-02	.2665722966010-02	.3769562619760-03
DELVI=	.3014774499490-02	.1814664421920-02	-.3504192294870-03

MINIMUM APPROACH DISTANCES

SUN	.691597[illegible]46009+00
MERCUR	.359b1A[illegible]757c2-06
VENUS	.44849794312bb-04
EARTH	.259416144106+04
MARS	.79485920176[illegible]+00
JUPITE	.45976046078[illegible]+01

FINAL APPROACH DISTANCES

SUN	.721531553080+00
MERCUR	.118469329906+01
VENUS	.44849794310[illegible]-04
EARTH	.924907463963+00
MARS	.219428134377+01
JUPITE	.571313574079+01

ROUTINE TERMINATED AT MERK$

RUN TERMINATED FOR MAX TIME AT 025644

B	031000 000000	000302 000032	000002 130723	000000 057603	000000 000160	000001 000007	000000 000000	000000 000000
	000000 000000	000000 171441	000000 163772	000000 003034	311622 120505	000000 000077	000000 000016	000000 000000
A	311622 120505	000000 000077	000000 000016	000000 000000	000000 000006	000000 000000	000000 000002	[illegible] 135012
	000001 400000	265377 714707	601715 371402	207521 530430	602325 241673	766632 510275	000000 000070	504703 504163
R	000000 000641	000000 026447	177447 124477	000000 000000	000000 000000	777777 777776	000000 035337	000000 000001
	000000 001323	000000 000000	000000 000000	000000 000000	777777 777776	777777 777776	777777 777776	000000 000021

TABLE V

Sample Data Set for NØPT=4 Case

The following data set will be assumed to be punched on cards started in column 1 with no spaces unless indicated.

```
Card 1--TP
D1=2.4443299842D+06,2.4445049474D+06,2.4446221842D+06
Card 2--XI
D2=-9.824794567024D-01,-1.820634010750D-01,9.417413113088D-06
Card 3--XF
D3=2.610368798355D-01,-4.199897439846D-01,-3.679846437010D-02
Card 4--VI
D4=2.691275379500D+00,-4.440086578800D+00,-1.322957073700D-01
Card 5--VF
D5=8.7038421339D+00,4.30945631490D+00,-6.446741114700D-01
Card 6--NP
I6=10,10
Card 7--NØPT
I7=4
Card 8--NPRT
I8=1
Card 9--NSPRT
I9=10$
```

Note that the dollar sign ($) should follow immediately the last digit of data in each data set in RDATA input.